

Competitive coevolutionary algorithm for robust multi-objective optimization: The worst case minimization

Ivan Reinaldo Meneghini

Graduate Program in Electrical Engineering
Federal University of Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brazil

Frederico Gadelha Guimarães

Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
Email: fredericoguimaraes@ufmg.br

Antonio Gaspar-Cunha

Institute for Polymers and Composites
University of Minho
Campus de Azurém
4800-058 Guimarães, Portugal
Email: agc@dep.uminho.pt

Abstract—Multi-Objective Optimization (MOO) problems might be subject to many modeling or manufacturing uncertainties that affect the performance of the solutions obtained by a multi-objective optimizer. The decision maker must perform an extra step of sensitivity analysis in which each solution should be verified for its robustness, but this post optimization procedure makes the optimization process expensive and inefficient. In order to avoid this situation, many researchers are developing Robust MOO, where uncertainties are incorporated in the optimization process, which seeks optimal robust solutions. We introduce a coevolutionary approach for robust MOO, without incorporating robustness measures neither in the objective function nor in the constraints. Two populations compete in the environment, one representing solutions and minimizing the objectives, another representing uncertainties and maximizing the objectives in a worst case scenario. The proposed coevolutionary method is a coevolutionary version of MOEA/D. The results clearly suggest that these competing co-evolving populations are able to identify robust solutions to multi-objective optimization problems.

Index Terms—Robust optimization, coevolutionary Algorithms, worst case minimization

I. INTRODUCTION

Engineering optimization problems are often characterized by multiple conflicting objective functions on many variables. The solutions to these problems form the so-called Pareto-optimal set, representing the tradeoff among the objectives. The corresponding image of this set in the objective space, the Pareto front, is characterized by nondominated points, i.e. improving one objective necessarily leads to deteriorating at least one of the other objectives. If the objectives admit a mathematical formulation, approximations of the optimal solutions can be achieved by means of multi-objective techniques. Among these, Evolutionary Algorithms (EA) [1, 2] present powerful results in many applications. These stochastic and bioinspired algorithms are able to find good approximations to the Pareto front in terms of convergence, while preserving diversity of the solutions in the estimate.

However, these problems may be affected by different types of uncertainties, such as noise, model inaccuracies, time variation, measurement imprecisions, disturbances and other uncontrolled effects, which might deteriorate the performance

of the designed solutions. The source of these uncertainties can be related to environmental variations (temperature, humidity, electromagnetic interference, etc), errors in sensors or measurement equipment, imprecisions in the manufacturing process or implementation of the numerical solution in the real world, simplifications or inaccuracies in the model behind the objective and constraint functions, or numerical or model approximations, since the objective functions can be approximated by means of experiments or interpolation (approximation) functions from data [3]. From the practical point of view, optimal solutions that are sensitive to perturbations of the variables or parameters of the model are not desirable. Identifying robust solutions in the estimate of the Pareto set returned by the optimization process is necessary but also can become an additional hindrance for the decision-maker, that should perform a sensitivity analysis of the solutions on the Pareto front in order to assess their robustness to these uncertainties. This analysis could be part of the decision-making process as additional criteria. An alternative to this post optimization procedure is to incorporate the robustness requirement in the optimization process, characterizing the Robust Multi-Objective Optimization (MOO) [4, 5]. The most used techniques in this area incorporate robustness in two different ways:

- Modifying the objective functions, using measures such as averages or deviations;
- Incorporating robustness measures into the problem as additional objectives or constraints.

Since these approaches incorporate the uncertainties of the problem in the objective function or in the constraint functions, it can be applied together with different metaheuristics, such as evolutionary algorithm and simulated annealing. Nevertheless, other interesting ideas and approaches have been presented in the literature, e.g. the use of co-evolutionary algorithms [6], the definition of new relations of dominance based on robustness [7, 8] or the use of interval analysis [9].

In this paper we introduce a coevolutionary approach for Robust MOO, without incorporating robustness measures nei-

ther in the objective function nor in the constraints. Coevolutionary algorithms are a class of evolutionary algorithms inspired by the simultaneous evolution of two or more interacting populations. Recently, various engineering problems have been solved with this approach [10, 11, 12, 13, 14]. Coello et al. [2] sets coevolution as a reciprocal evolutionary change between species that interact with each other. Coevolutionary algorithms can be categorized into competitive or cooperative algorithms, depending on the nature of the interaction between individuals. In a competition relationship, both species have a negative effect on each other since they are competing for the same resources. In the competitive approach, individuals in the populations compete among themselves, characterizing an *arms race* or the classical *predator-prey* coevolution. Cooperative coevolution is characterized by a beneficial interaction between individuals from each population. Typically, cooperative coevolution is used in decomposition of problems into subproblems, and each population evolves partial solutions to each subproblem [13]. The cooperation is required to compute the true fitness of a solution.

In the proposed competitive approach, we create two populations that compete in the environment, one representing the design variables and another representing uncertainties. These populations interact in the computation of the fitness values. The population of variables is evolving towards the Pareto front considering the uncertainties of the second population, while the population of uncertainties is evolving towards deteriorating the estimate of the Pareto front through perturbations to the variables or to parameters of the objective functions. The proposed co-evolutionary method is a co-evolutionary version of MOEA/D [15]. The results presented in the experiments clearly suggest that these competing co-evolving populations are able to identify robust solutions into multi-objective optimization problems.

This work is organized as follows. Section II is a review of robust multi-objective concepts, multi-objective evolutionary algorithm and co-evolutionary algorithms. Special attention is given to the MOEA/D algorithm. Section III presents the proposed competitive coevolutionary algorithm, a coevolutionary version of MOEA/D developed for worst case minimization. Results and discussion are presented in Section IV.

II. BACKGROUND

A. Robust Optimization

An important reference in robust design is the work developed by the Japanese engineer and statistician Genichi Taguchi [16], with influential contributions to industrial statistics and process quality engineering. According to Taguchi [16], quality is measured by the deviation that a functional characteristic presents with respect to its expected value. In this way, a process is said to be robust when it is less sensitive to disturbances in its parameters. Zang et al. [17] defined signal factors as the parameters of a process that determine the configurations to be considered in the robust design, noise factors as factors that are hard to control, causing variability in the system, and control factors as those factors that should

be optimized in order to reduce the sensitivity of the system to the noise factors. Consider $f(s, z, x)$ as a response vector for a determined set of values, where s , z and x represent respectively the signal factors, the noise factors and the control factors. The mathematical formulation presented by Zang et al. [17] for the robust design is given by:

$$\hat{x} = \arg \min_x \left(\max_{s \in V} E_z [|f(s, z, x) - t|^2] \right) \quad (1)$$

where t was the desired response and E was the statistical expectancy (in z), hence characterizing the robust design as a *min-max* problem.

Later, Deb and Gupta [18] classified the robustness of a solution according to the way the optimization problem was formulated. In the case of single objective optimization, i.e. the minimization of a function $f(x) \in \mathbb{R}$ with $x \in S$:

- A robust solution of **Type I** is obtained through the minimization of the mean value of $f(x)$ in the defined neighborhood $B_\delta(x)$. In other words, a robust solution is characterized by formulating the problem as the minimization of a mean effective function

$$f^{eff} = \frac{1}{|B_\delta(x)|} \int_{y \in B_\delta(x)} f(y) dy,$$

where $|B_\delta(x)|$ is the hypervolume of the neighborhood $B_\delta(x)$.

- A robust solution of **Type II** is obtained when the perturbation is considered by adding constraints to the problem, which is formulated as the minimization of the function $f(x)$ subject to

$$\frac{||f^p(x) - f(x)||}{||f(x)||} \leq \eta,$$

where the perturbed function $f^p(x)$ can be the mean effective function $f^{eff}(x)$ or the worst case value observed in the neighborhood $B_\delta(x)$.

In the same paper, Deb and Gupta [18] extended these definitions to multi-objective problems, also presenting benchmark functions for Robust MOO.

Jin and Branke [3] presented a classification of sources of uncertainties in optimization problems, based on how these affect the optimization process:

- Presence of noise in the evaluation of objective functions;
- Sensitivity of the solution (in the objective space) to perturbations in the variables (design space). Those solutions that are immune to this are called robust solutions;
- Fitness function approximations, in cases where the function is expensive to evaluate or the analytical expression is not available;
- Time variations of the objective functions, such that the function, and accordingly its minima, change with time.

The authors provided an overview of the main techniques utilized in each case, for instance, implicit or explicit means of the functions, modifications in selection and self-adaptation in evolutionary algorithms, and others [3].

Goh et al. [5] investigated problems with the presence of noise and robust optimization problems, classifying the ways how robustness measures are included in the optimization process as: (i) incorporating noise and/or robustness measures in the variables of the problem as in [4, 19]; (ii) considering these factors as additional objectives to be optimized.

In addition to these methodologies, other ideas had been presented in the literature. Li et al. [6] presented a coevolutionary algorithm for the robust design of a *permanent magnet machine*, considering the worst case minimization. The worst case minimization consists in the following optimization problem:

$$\min_{x \in X} \left\{ \max_{u \in U} f(x, u) \right\} \quad (2)$$

in which $f(x, u)$ is the objective function, x represents the design variables, u represents the uncertainties. However, their coevolutionary approach is limited to single objective optimization.

Soares et al. [9] presented an interval based multi-objective evolutionary algorithm for robust optimization (**IJMOEA**), assuming robustness as insensitivity to variable perturbations. The uncertain parameters are represented with intervals, which results in solution objectives also being represented with intervals. The comparison of solutions is done in the worst-case scenario values of objectives, that is, the values at the border of an interval.

B. Multi-objective optimization

A constrained multi-objective optimization problem (MOP) can be defined as [1, 15]

$$\left. \begin{array}{l} \text{Minimize } F(x) = (f_1(x) \dots f_M(x)) \\ \text{subject to } x \in X \subset \mathbb{R}^n \\ g_j(x) \geq 0, \quad 1 \leq j \leq J \\ h_k(x) = 0, \quad 1 \leq k \leq K \end{array} \right\} \quad (3)$$

where X is the decision or design space, $F(X) \subset \mathbb{R}^M$ is the objective space, $f_i(x)$ are the objective functions, $g_j(x)$, $j = 1, \dots, J$ are the inequality constraints, and $h_k(x)$, $k = 1, \dots, K$ are the equality constraints. The constraints of the problem define the feasible region in the decision space and the corresponding image in the objective space.

Multi-objective Evolutionary Algorithms (MOEA) [1, 18, 15] have been successful methods to obtain an approximation of the Pareto front of a MOP. These techniques have been applied in many different fields. In MOEA, new solutions are generated at every iteration from the application of selection and genetic operators on an existing population. The performance of solutions is assessed with a fitness function or comparison procedures. Usually, an external population preserves the best estimate of the Pareto front found so far and it is updated accordingly [1].

However, MOEA do not guarantee the robustness of the solutions, which is desirable in many practical applications,

given the presence of uncertainties. Robust optimization seeks for feasible and efficient solutions, that would be less sensitive to perturbations to the variables or parameters of the objective functions [20, 21, 22, 23].

C. MOEA/D

An efficient strategy used in some MOEA is decomposition, in which the MOP is divided into scalar optimization subproblems. One important example of algorithm following this strategy is MOEA/D, proposed by Zhang and Li [15]. In MOEA/D, a set of vectors $\Lambda = \{\lambda^i \in F(X); 1 \leq i \leq P\}$, with population size P , is generated before the optimization. These vectors will direct the optimization process. Each individual of the population is associated to a vector λ^i , which is minimized according to one of the methods below:

- **Weighted Sum:** Considers a convex combination of the objectives, which should be minimized in the scalar problem defined by:

$$\text{minimize } g^{ws}(x|\lambda) = \sum_{i=1}^M \lambda_i f_i(x) \quad (4)$$

with $\lambda = (\lambda_1, \dots, \lambda_M)$ and $\sum_{j=1}^M \lambda_j = 1$

- **Tchebycheff:** Considers the minimization of the scalar problem defined by

$$\text{minimize } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq M} \{\lambda_i |f_i(x) - z_i^*|\} \quad (5)$$

with $\lambda = (\lambda_1, \dots, \lambda_M)$ and $z^* = (z_1^*, \dots, z_M^*)$, $z_i^* = \min f_i(x)$.

- **Boundary Intersection:** Considers the minimization of the scalar problem defined by

$$\text{minimize } g^{bi}(x|\lambda, z^*) = d_1 + \theta d_2 \quad (6)$$

with $d_1 = \frac{\|(F(x) - z^*)^T \lambda\|}{\|\lambda\|}$, $d_2 = \|F(x) - (z^* + d_1 \lambda)\|$ and $\theta > 0$ is a penalty factor.

In each of the strategies above, the MOP is decomposed into the simultaneous minimization of scalar subproblems defined by g , in which the search is restricted to a group of individuals in the neighborhood $B_\epsilon(x)$ associated with an element $\lambda^i \in \Lambda$.

The MOEA/D is summarized in the following steps:

1) Initialization:

- Create an empty external population EP, which stores the estimates of the Pareto front;
- Generate initial population $X = \{x^1, \dots, x^P\}$, with P individuals;
- Generate P weight vectors Λ for the scalar subproblems;
- Calculate the Euclidean distance between any two weight vectors and, for each λ^i , find the T closest vectors. Create a neighborhood $B(i) = \{i_1, \dots, i_T\}$ referring to $\lambda^{i_1}, \dots, \lambda^{i_T}$, which are the T closest weight vectors to λ^i . Associate each individual to a single weight vector according to the distance in the objective space or randomly.

- e) Evaluate X , making $FV^i = F(x^i)$;
 - f) Initialize the estimated utopian solution $z = (z_1, \dots, z_M)$, with a problem-specific method. Here we set $z_i = \min f_i(x^j)$ with $x^j \in X$.
- 2) **Evolutionary cycle:** In each iteration, for each individual x^i do:
- a) Select two indices at random k, l from $B(i)$. Generate new offspring x_c with the genetic operators.
 - b) Update $z = (z_1, \dots, z_m)$: if $z_j > f_j(x_c)$ for any $j = 1 \dots m$ then do $z_j = f_j(x_c)$.
 - c) Update the neighboring solutions: for each $j \in B(i)$, if $g(x_c | \lambda^j, z) \leq g(x^j | \lambda^j, z)$ then set $x^j = x_c$ and $FV^j = F(x_c)$.
 - d) Update EP .

The reference point z in 2b is related to the scalar subproblems. In minimization problems, this point corresponds to the smallest value in all objectives, that is, $z = (z_1, \dots, z_m)$ with $z_i = \min f_i(x)$.

Note that each individual in the population is evolving to minimize one sub-problem. The minimization of each sub-problem can potentially lead to a Pareto-optimal solution. The search process uses information from solutions to those similar sub-problems, those that belong to the neighborhood $B(i)$ of x^i .

III. COEVOLUTIONARY ROBUST MOEA/D (C-RMOEA/D)

The idea behind the proposed method, called Coevolutionary Robust MOEA/D (C-RMOEA/D), is the employment of a coevolutionary approach for the solution of robust MOP considering the worst case minimization. The method is based on the basic MOEA/D but under a coevolutionary scheme. The choice for MOEA/D is explained by its characteristic decomposition of the multi-objective problem into scalar subproblems. The decomposition facilitates the implementation of a coevolutionary approach as it will be shown below.

At each iteration of the evolutionary cycle, two instances of optimization problems are concurrently approached. The population of individuals representing the candidate solutions, $X = \{x^1, \dots, x^P\}$, is evolving towards the minimization of the function g given the values of perturbations in the population Δ . On the other hand, the population Δ is evolving towards the maximization of the function g given the values in X . These two processes are in conflict or competition with one another. In other words, the population X , related to the design variables, and the perturbation population Δ compete in the environment defined by the robust MOP, see the flowchart in Fig. 1 for details.

Therefore, C-RMOEA/D works with two populations, X , representing the candidate solutions for the design variables, and Δ , representing perturbations added to the design variables and/or perturbations in parameters of the functions, both with size P . For each $x^i = (x_1^i, \dots, x_n^i) \in X$ there is a corresponding $\delta^i = (\delta_1^i, \dots, \delta_q^i) \in \Delta$ with $\delta_j^i \in [-\epsilon, \epsilon]$.

After the initialization steps of C-RMOEA/D, the evolutionary cycle begins for both populations. For each $x^i \in X$,

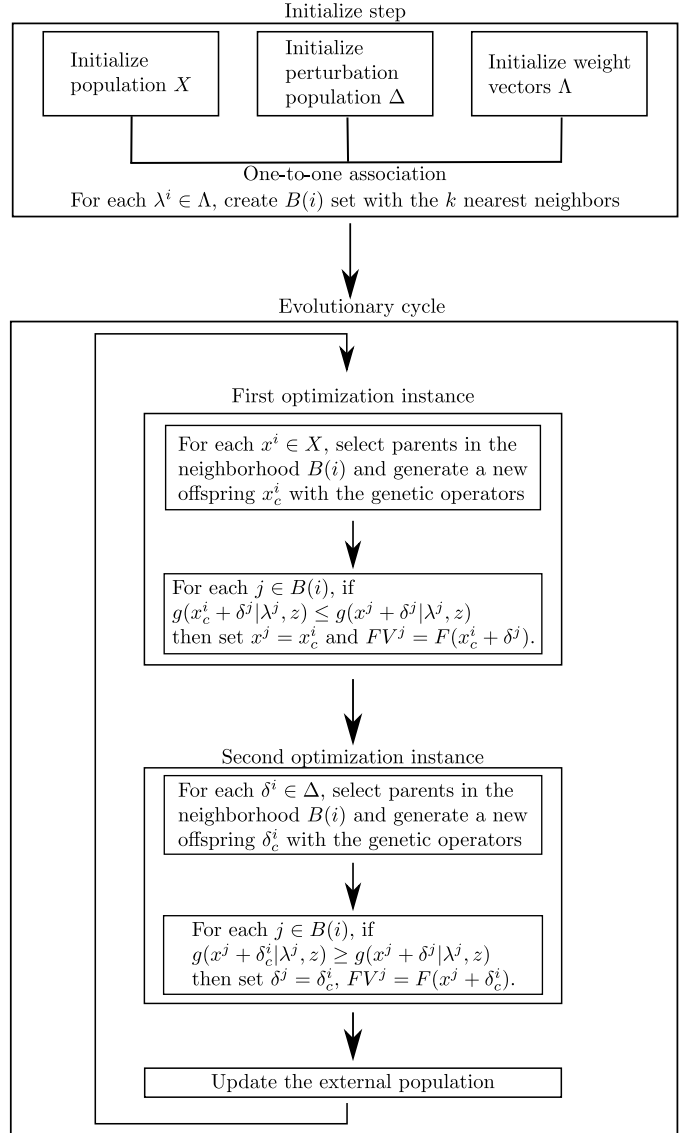


Fig. 1. The two instances of the worst case robust optimization: at each iteration, the design variables and the uncertainty perturbations are evolved causing the minimization of the function g in one direction and the maximization in the opposite direction. These two processes are in conflict or competition with one another. In other words, the population X , related to the design variables, and the perturbation population Δ compete in the environment defined by the robust MOP.

two neighboring individuals x_a^i and x_b^i are selected in the neighborhood $B(i)$. After that, an offspring is created with the application of the genetic operators (crossover and mutation), producing a new solution x_c^i . In this case, we are trying to minimize the subproblem given by:

$$\min_x g(x | \delta^i, \lambda^i, z) \quad (7)$$

Notice that in (7), we use the perturbation vector δ^i . Therefore, we should update the neighboring solutions as follows: for each $j \in B(i)$, if

$$g(x_c^i | \delta^j, \lambda^j, z) \leq g(x^j | \delta^j, \lambda^j, z)$$

then set $x^j = x_c^i$ and $FV^j = F(x_c^i, \delta^j)$.

Analogously, for each $\delta^i \in \Delta$, two neighboring individuals δ_a^i and δ_b^i are selected in the neighborhood $B(i)$. An offspring δ_c^i is produced, which is related to the maximization of the g function:

$$\max_{\delta} g(\delta | x^i, \lambda^i, z) \quad (8)$$

In (8), we use the variable x^i in order to evaluate δ . Similarly, we update the neighboring vectors as follows: for each $j \in B(i)$, if

$$g(\delta_c^i | x^j, \lambda^j, z) \geq g(\delta^j | x^j, \lambda^j, z)$$

then set $\delta^j = \delta_c^i$ and $FV^j = F(x^j, \delta_c^i)$.

Notice that one population is minimizing g given the values of the perturbations, while the other population is maximizing g given the values of the variables. These two competing process happen for each pair of $x^i \in X$ and $\delta^i \in \Delta$ and for each $\lambda^i \in \Lambda$. The decomposition approach in MOEA/D makes the coevolutionary approach simpler, since we evolve the perturbations and the variables considering each subproblem independently.

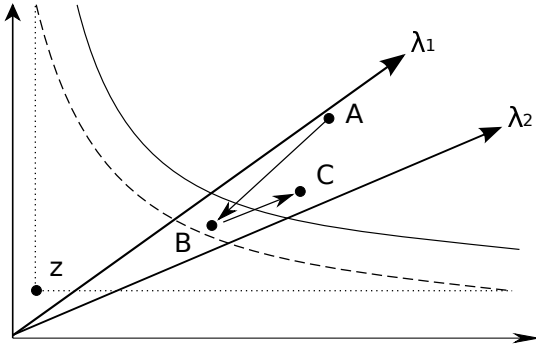


Fig. 2. Illustration of the competitive coevolution in C-RMOEA/D.

Fig. 2 illustrates the competitive coevolutionary process in the objective space. After an offspring x_c^i is created, point A is updated to point B, which decreases the value of g given λ^1, z . After that, in the evolutionary cycle of Δ , it might happen that the point B is updated to point C, which increases the value of g given λ^1, z . This alternating process lead to a robust estimate of the Pareto front, considering the worst case analysis. The estimated front contains those solutions that minimize the objectives in the worst case of the perturbations. In one evolutionary cycle, the individuals in X are adjusted according to (7). The estimated Pareto front moves towards the utopian solution in the direction of the front shown in green line. In the evolution of Δ , the perturbations are adjusted according to (8). The estimated Pareto front moves away from the utopian solution, in the direction of the front shown in blue line, because the new perturbations found increase the objective function values. By repeating this process, until convergence criteria are met, we can find an estimate of the robust Pareto front, i.e., an estimate that is good even considering the worst case scenario of perturbations for each specific point.

Given that C-RMOEA/D is just an extension of MOEA/D, its implementation is very simple, as presented below.

1) Initialization:

- Create an empty external population EP, which stores the estimates of the Pareto front;
- Generate initial population $X = \{x^1, \dots, x^P\}$, with P individuals;
- Generate population $\Delta = \{\delta^1, \dots, \delta^P\}$, with P individuals;
- Associate each individual in X to an individual in Δ .
- Generate P weight vectors Λ for the scalar sub-problems;
- Calculate the Euclidean distance between any two weight vectors and, for each λ^i , find the T closest vectors. Create a neighborhood $B(i) = \{i_1, \dots, i_T\}$ referring to $\lambda^{i_1}, \dots, \lambda^{i_T}$, which are the T closest weight vectors to λ^i . Associate each individual to a single weight vector according to the distance in the objective space or randomly.
- Evaluate X , making $FV^i = F(x^i, \delta^i)$;
- Initialize the estimated utopian solution $z = (z_1, \dots, z_m)$, with a problem-specific method. Here we set $z_i = \min f_i(x^j, \delta^j)$ with $x^j \in X$.

2) Evolutionary cycle:

- For each individual x_i do:
 - Generate an offspring x_c^i with the genetic operators.
 - Update the neighboring solutions: for each $j \in B(i)$, if

$$g(x_c^i | \delta^j, \lambda^j, z) \leq g(x^j | \delta^j, \lambda^j, z) \quad (9)$$

then set $x^j = x_c^i$ and $FV^j = F(x_c^i, \delta^j)$.

- For each $\delta^i \in \Delta$ do:
 - Generate an offspring δ_c^i with the genetic operators.
 - Update the neighboring solutions: for each $j \in B(i)$, if

$$g(\delta_c^i | x^j, \lambda^j, z) \geq g(\delta^j | x^j, \lambda^j, z) \quad (10)$$

then set $\delta^j = \delta_c^i$, $FV^j = F(x^j, \delta_c^i)$.

- Update EP.

IV. RESULTS AND DISCUSSION

A. Test problems

In order to verify the efficacy of C-RMOEA/D, the benchmark functions ZDT1 and ZDT2 proposed by Zitzler et al. [24] and TP2, TP2 and TP4 proposed by Gaspar-Cunha et al. [25] were used in this study. In all these functions the perturbation vector is $\delta^i = (\delta_1^i, \dots, \delta_q^i) \in \Delta$ with $\delta_j^i \in [-0.025, 0.025]$. The choice of ϵ is a parameter defined by the Decision Maker and $\epsilon = 0.025$ makes the algorithm search for robust solutions in a range of length $2\epsilon = 0.05$ around the optimal solutions by the worst case minimization approach. Considering the

size of the variable range ($x_i \in [0, 1]$ in all problems), the obtained solutions sets the trade off between the highest value of the goals at an interval of 5% of the range of the variables around the optimal value x^* . As the first objective in ZDT1 and ZDT2 functions is $f_1(x, \delta) = x_1 + \delta_1$, the uncertainty of x_1 is transmitted to this goal, establishing an uncertainty in the objective space.

• **ZDT1:**

$$\min_{x \in [0,1]} \max_{\delta \in [-0.025, 0.025]} \begin{cases} f_1(x, \delta) = x_1 + \delta_1 \\ f_2(x, \delta) = g \times h \end{cases} \quad (11)$$

where:

$$g = 1 + \frac{9}{n-1} \sum_{i=2}^n (x_i + \delta_i)$$

$$h = 1 - \sqrt{f_1/g}$$

• **ZDT2:**

$$\min_{x \in [0,1]} \max_{\delta \in [-0.025, 0.025]} \begin{cases} f_1(x, \delta) = x_1 + \delta_1 \\ f_2(x, \delta) = g \times h \end{cases} \quad (12)$$

where:

$$g = 1 + \frac{9}{n-1} \sum_{i=2}^n (x_i + \delta_i)$$

$$h = 1 - \left(\frac{f_1}{g} \right)^2$$

• **TP2:**

$$\min_{x \in [0,1]} \max_{\delta \in [-0.025, 0.025]} \begin{cases} f_1(x, \delta) = \cos \left(\frac{\pi(x_1 + \delta_1)}{2} \right) \\ f_2(x, \delta) = g \sin \left(\frac{\pi(x_1 + \delta_1)}{2} \right) \end{cases} \quad (13)$$

where:

$$g = 1 + \frac{10}{n-1} \left(\sum_{i=2}^n (x_i + \delta_i) \right)$$

• **TP3:**

$$\min_{x \in [0,1]} \max_{\delta \in [-0.025, 0.025]} \begin{cases} f_1(x, \delta) = 1 - (x_1 + \delta_1)^2 \\ f_2(x, \delta) = g \sin \left(\frac{\pi(x_1 + \delta_1)}{2} \right) \end{cases} \quad (14)$$

where:

$$g = 1 + \frac{10}{n-1} \left(\sum_{i=2}^n (x_i + \delta_i) \right)$$

• **TP4:**

$$\min_{x \in [0,1]} \max_{\delta \in [-0.025, 0.025]} \begin{cases} f_1(x, \delta) = \frac{e^{(x_1 + \delta_1)} - 1}{e - 1} \\ f_2(x, \delta) = g \cdot \left[\frac{\sin(4\pi(x_1 + \delta_1)) - 15(x_1 + \delta_1)}{15} + 1 \right] \end{cases} \quad (15)$$

where:

$$g = 1 + \frac{10}{n-1} \left(\sum_{i=2}^n (x_i + \delta_i) \right)$$

The Pareto front for these benchmark functions is known analytically, thus it is possible to obtain an estimate of the robust front by Monte Carlo simulation over the points on the true Pareto front. This is done by sampling with a Uniform distribution under a Latin Square different values of perturbations for each solution on the front and re-evaluating the objective functions. The worst case corner estimated with Monte Carlo provides an estimate of the robust front.

In our experiments, C-RMOEA/D was run with the following configuration: 300 iterations, population size of $P = 210$ ¹ for both populations, crossover probability $pc = 1$ and mutation probability $pm = \frac{1}{n}$. Each problem was set with 10 design variables, i.e., $n = 10$. Figures 3, 4, 5, 6 and 7 illustrate the results achieved by C-RMOEA/D.

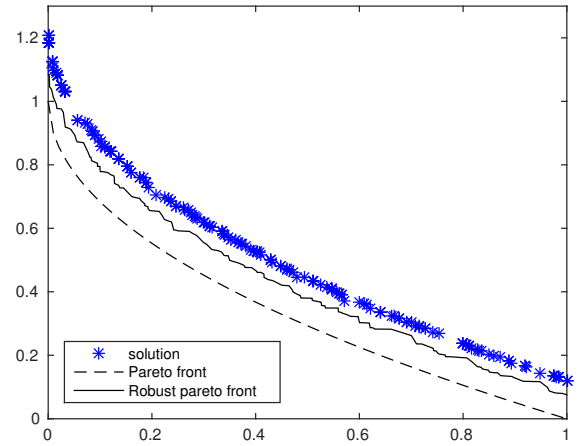


Fig. 3. Test problem ZDT1

These figures show the non robust Pareto front of the original problem, a worst case estimate applied on the non robust front with Monte Carlo simulation, and the dots represent the solutions achieved by the proposed method. These preliminary results are encouraging evidence that the proposed C-RMOEA/D is able to achieve approximations of the robust front, without requiring additional function evaluations in sampling nor any robust measure of the objective functions. The Pareto front for all problems correspond to $x_1 \in [0, 1]$ and $x_i = 0$ for $i > 1$.

B. Conclusions

Different types of uncertainties, such as noise, model inaccuracies, time variation, measurement imprecisions, disturbances and other uncontrolled effects, are always present in real applications, affecting the nominal performance of a design. Directly considering them in the optimization process

(15)¹ According to the number of weight vectors generated.

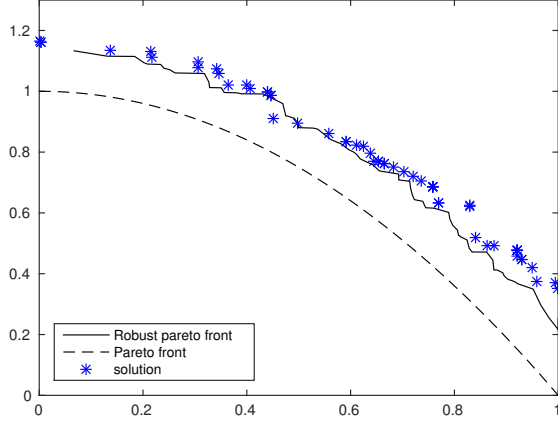


Fig. 4. Test problem ZDT2

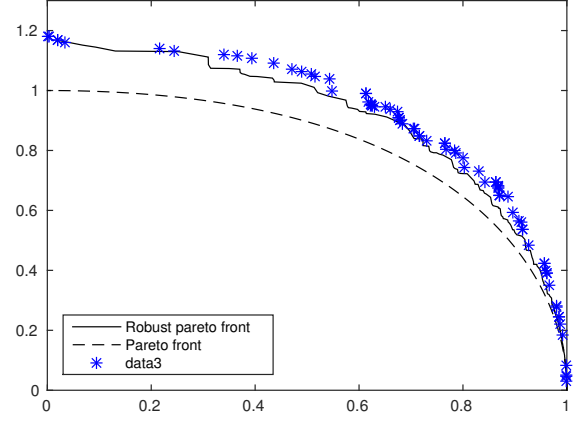


Fig. 6. Test problem TP3

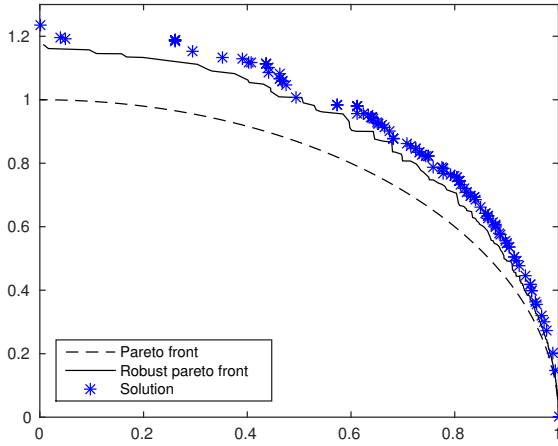


Fig. 5. Test problem TP2

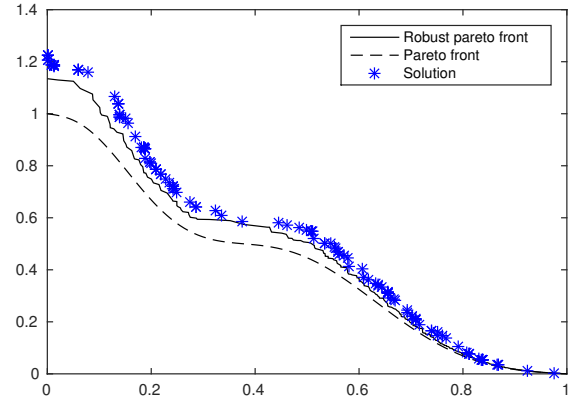


Fig. 7. Test problem TP4

lead to the Robust MOO, bringing benefits to the design process. However, usual optimization techniques should be modified and adapted to incorporate robustness requirements into the problem.

This work considered the worst case scenario for the multi-objective optimization problem. A competitive coevolutionary approach based on MOEA/D was introduced for the solution of robust MOP. The algorithm works with two competing populations, one representing the design variables and another representing perturbations added to the design variable or perturbations in parameters of the objective functions. The population of individuals representing the candidate solutions X is evolving towards the minimization of the function g given the values of perturbations in the population Δ . On the other hand, the population Δ is evolving towards the maximization of the function g given the values in X . These two processes are in conflict or competition with one another.

The results reported in this paper represent preliminary

experiments with the proposed method. These results show the adequacy and efficacy of the method, being able to converge to an estimate of the robust Pareto front. Since two populations are considered in the coevolutionary method, the number of objective function evaluations is doubled in comparison to a single population algorithm. Nonetheless, this additional cost can be small compared to other strategies based on fitness sampling and robust measures. The proposed method is promising in the context of robust multi-objective optimization, in addition to being an original way of approaching the problem.

Acknowledgements

This work has been supported by the Brazilian agency CAPES. The authors also would like to thank Brazilian agency CNPq (grants 306694/2013-1) and FAPEMIG, for supporting the development of this work.

The first author would like to thank the support given by Instituto Federal Minas Gerais (IFMG), Brazil.

The third author wants to acknowledge the support of FEDER funds through the COMPETE 2020 Programme and

REFERENCES

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [2] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [3] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments – a survey," *Trans. Evol. Comp.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.
- [4] A. Gaspar-Cunha and J. Covas, "Robustness in multi-objective optimization using evolutionary algorithms," *Computational Optimization and Applications*, vol. 39, no. 1, pp. 75–96, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s10589-007-9053-9>
- [5] C. K. Goh, K. C. Tan, C. Y. Cheong, and Y. S. Ong, "An investigation on noise-induced features in robust evolutionary multi-objective optimization," *Expert Syst. Appl.*, vol. 37, no. 8, pp. 5960–5980, Aug. 2010.
- [6] M. Li, F. G. Guimarães, and D. A. Lowther, "Competitive co-evolutionary algorithm for constrained robust design," *IET Science, Measurement & Technology*, vol. 9, pp. 218–223(5), March 2015.
- [7] M. Mlakar, T. Tušar, and B. Filipič, "Comparing solutions under uncertainty in multiobjective optimization," *Mathematical Problems in Engineering*, vol. 2014, 2014, article ID 817964.
- [8] M. Li, R. P. Silva, F. G. Guimarães, and D. Lowther, "A new robust dominance criterion for multiobjective optimization," *IEEE Transactions on Magnetics*, vol. 51, no. 3, p. 14, February 2015.
- [9] G. L. Soares, F. G. Guimarães, C. A. Maia, J. A. Vasconcelos, and L. Jaulin, "Interval robust multi-objective evolutionary algorithm," in *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation*, ser. CEC'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1637–1643.
- [10] I. Blecic, A. Cecchini, and G. A. Trunfio, "Fast and accurate optimization of a gpu-accelerated {CA} urban model through cooperative coevolutionary particle swarms," *Procedia Computer Science*, vol. 29, pp. 1631 – 1643, 2014, 2014 International Conference on Computational Science.
- [11] H. Chen, Y. Mori, and I. Matsuba, "Solving the balance problem of massively multiplayer online role-playing games using coevolutionary programming," *Appl. Soft Comput.*, vol. 18, pp. 1–11, May 2014.
- [12] A. Ladjici, A. Tiguercha, and M. Boudour, "Nash equilibrium in a two-settlement electricity market using competitive coevolutionary algorithms," *International Journal of Electrical Power & Energy Systems*, vol. 57, pp. 148 – 155, 2014.
- [13] F. B. de Oliveira, R. Enayatifar, H. J. Sadaei, F. G. Guimarães, and J.-Y. Potvin, "A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem," *Expert Syst. Appl.*, vol. 43, no. C, pp. 117–130, Jan. 2016.
- [14] J. Branke and J. Rosenbusch, *Parallel Problem Solving from Nature – PPSN X: 10th International Conference, Dortmund, Germany, September 13-17, 2008. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. New Approaches to Coevolutionary Worst-Case Optimization, pp. 144–153.
- [15] Q. Zhang and H. Li, "A multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, *Accepted*, vol. 2007, 2007.
- [16] G. Taguchi, *Introduction to quality engineering: designing quality into products and processes*. Quality Resources, 1986.
- [17] C. Zang, M. I. Friswell, and J. E. Mottershead, "A review of robust optimal design and its application in dynamics," *Comput. Struct.*, vol. 83, no. 4-5, pp. 315–326, Jan. 2005.
- [18] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization," *Evol. Comput.*, vol. 14, no. 4, pp. 463–494, Dec. 2006. [Online]. Available: <http://dx.doi.org/10.1162/evco.2006.14.4.463>
- [19] J. Ferreira, C. M. Fonseca, J. A. Covas, and A. Gaspar-Cunha, *Evolutionary Multi-Objective Robust Optimization, Advances in Evolutionary Algorithms*, X. Zhihui, Ed. InTech, 2008.
- [20] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton Series in Applied Mathematics, 2009.
- [21] D. Pflugfelder, J. J. Wilkens, and U. Oelfke, "Worst case optimization: a method to account for uncertainties in the optimization of intensity modulated proton therapy," *Physics in Medicine and Biology*, vol. 53, no. 6, p. 1689, 2008.
- [22] W. Liu, X. Zhang, Y. Li, and R. Mohan, "Robust optimization of intensity modulated proton therapy," *Medical Physics*, vol. 39, no. 2, pp. 1079–1091, February 2012.
- [23] A. Fredriksson and R. Bokrantz, "A critical evaluation of worst case optimization methods for robust intensity-modulated proton therapy planning," *Medical Physics*, vol. 41, no. 8, 2014.
- [24] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [25] A. Gaspar-Cunha, J. Ferreira, and G. Recio, "Evolutionary robustness analysis for multi-objective optimization: benchmark problems," *Structural and Multidisciplinary Optimization*, vol. 49, no. 5, pp. 771–793, 2013.